Related Randomness Attacks for Public Key Encryption

Kenneth G. Paterson, Jacob C. N. Schuldt, Dale L. Sibborn *

Information Security Group, Royal Holloway, University of London
{kenny.paterson,jacob.schuldt,dale.sibborn.2011}@rhul.ac.uk

Abstract. Several recent and high-profile incidents give cause to believe that randomness failures of various kinds are endemic in deployed cryptographic systems. In the face of this, it behoves cryptographic researchers to develop methods to immunise – to the extent that it is possible – cryptographic schemes against such failures. This paper considers the practically-motivated situation where an adversary is able to force a public key encryption scheme to reuse random values, and functions of those values, in encryption computations involving adversarially chosen public keys and messages. It presents a security model appropriate to this situation, along with variants of this model. It also provides necessary conditions on the set of functions used in order to attain this security notation, and demonstrates that these conditions are also sufficient in the Random Oracle Model. Further standard model constructions achieving weaker security notions are also given, with these constructions having interesting connections to other primitives including: pseudo-random functions that are secure in the related key attack setting; Correlated Input Secure hash functions; and public key encryption schemes that are secure in the auxiliary input setting (this being a special type of leakage resilience).

1 Introduction

Modern cryptographic primitives are heavy consumers of randomness. Unfortunately, random number generators (RNGs) used to provide this randomness often fail in practice [16, 18, 20, 21, 13, 1, 15, 26]. This is due to issues including poor algorithmic design, software bugs, insufficient or poor estimation of system entropy, and the handling of randomness across virtual machine resets [27]. The results of randomness failures can be catastrophic and newsworthy in practice – DSA, ECDSA and Schnorr private signing keys can be exposed [9, 27]; plaintext recovery for low entropy plaintext becomes possible in the the public key encryption setting; key generation processes can be severely weakened [13, 24, 22, 10]; ephemeral Diffie-Hellman keys can become predictable leading to compromise of session keys [18]; and electronic wallet security can be compromised [11].

Evidently, randomness failures are a major problem in practice. The cryptography research community has begun to address this problem only relatively

* All authors were supported by EPSRC Leadership Fellowship EP/H005455/1.

recently [28, 29, 23, 2, 33, 27]. Accepting that randomness failures are endemic and unlikely to be eliminated in totality, a basic approach is to try to *hedge* against randomness failures, that is, to design cryptographic primitives that still offer a degree of security in the face of randomness failures. For signatures, there is a folklore de-randomisation technique which neatly sidesteps security issues arising from randomness failures: simply augment the signature scheme's private key with a key for a pseudo-random function (PRF), and derive any randomness needed during signing by applying this PRF to the message to be signed; meanwhile verification proceeds as normal. In the symmetric encryption setting, previous work has considered nonce-based encryption [28], misuse-resistant authenticated encryption (which concerns residual security when nonces are repeated) [29], and encryption in a chosen-randomness setting (wherein the adversary is given control over the randomness used for encryption) [23]. Ristenpart and Yilek [27] studied the use of "hedging" as a general technique for protecting against broad classes of randomness failures in already-deployed systems, and implemented and benchmarked this technique in OpenSSL. Hedging in the sense of [27] involves replacing the random value r required in some cryptographic scheme with a hash of r together with other contextual information, such as a message, algorithm or unique operation identifier, etc. Their results, while applying to a variety of different randomness failure types (see in particular [27, Figure 3]), all have their security analyses restricted to the ROM. Work in the public key encryption setting can be summarised as follows:

- Bellare *et al.* [2] considered security under *chosen distribution attack*, wherein the joint distribution of message and randomness is specified by the adversary, subject to containing a reasonable amount of min entropy. The PKE scheme designer's challenge is to find a way of "extracting" this entropy in a secure way. Bellare *et al.* gave several designs for PKE schemes achieving this notion in the Random Oracle Model (ROM) and in the standard model. This is a powerful and general approach, but does have its limitations: under extreme failure conditions, the joint message-randomness distribution may simply *fail* to contain sufficient entropy, at which point all security guarantees may be lost; moreover, for technical reasons, the model in [2] requires the target public key to be hidden from the adversary until all encryption queries have been made. This is impractical in real world applications.
- Yilek [33], inspired by virtual machine reset attacks in [27], considered the scenario where the adversary does not know the randomness (in contrast to the chosen-randomness setting of [23]), but can instead force the reuse of random values that are otherwise well-distributed. This is referred to in [33] as the *Reset Attack* (RA) setting. To fully reflect the reality of randomness failures in this setting, Yilek provides the adversary with the ability to encrypt chosen messages under adversarially generated public keys using the unknown but repeated random values. This makes his model very powerful, to the extent that certain trivial attacks must be excluded by assuming the adversary is *equality-pattern respecting*. In [33], Yilek also gave a general construction in which the random coins of the encryption algorithm are used

as a key to a PRF, the input to the PRF is the public key concatenated with the message to be encrypted, and the output of the PRF is then used as the 'randomness' for the encryption algorithm. This is sufficient to achieve security in his RA setting. Note that the RA security model is incomparable with the CDA model of [2].

1.1 Motivation

Inspired by the challenge of preserving security under randomness failures, we initiate the study of security for PKE in what we call the *Related Randomness Attack* (RRA) setting. Our RRA setting builds on the RA setting from [33] and brings the theory of hedging PKE against randomness failures closer to practice. As we shall see, it also has interesting connections with related key attacks for PRFs and PKE, as developed in [5, 3, 4, 6, 32], and leakage resilient cryptography (and in particular, the techniques developed in [14] to provide security for PKE in the auxiliary input setting).

In our RRA setting, the adversary can now not only force the reuse of existing random values as in the RA setting, but can also force the use of *functions of* those random values. This power is analogous to the power granted to the adversary in the Related Key Attack (RKA) setting, wherein an adversary is able to tamper with private (or secret) keys used during cryptographic operations. The RA setting arises as the special case of our RRA setting where only the identity function is allowed. The extra adversarial power in the RRA setting allows the modelling of reset attacks in which the adversary does not have an exact reset capability, but where the randomness used after a reset is in some way related to that used on previous resets. Such behaviours were observed in the experimental work in [27]. Furthermore, our RRA setting allows modelling of situations where the randomness used in a scheme comes from a PRNG which is not regularly refreshed with new entropy, but which steps forward under some deterministic state evolution function $\texttt{Next}$ and output function $\texttt{Out}$; here the appropriate functions in our RRA setting would be the compositions $\texttt{Out}(\texttt{Next}^i(\cdot))$.

More generally, RRA security has a strong theoretical motivation as being a stepping stone towards giving the adversary enhanced control over the inputs to cryptographic algorithms – messages (in the standard PKE setting), keys (in the RKA setting), and now randomness (in our new RRA setting). It is an interesting direction for future research to develop this theme further, by examining security in a combined RKA/RRA setting, where the adversary would be able to simultaneously tamper with *all* the inputs to a PKE scheme.

1.2 Our contributions

RRA security model In this paper, we provide a strong model and security definition for PKE in the RRA setting, which we name RRA-ATK security (where ATK = CPA or CCA) . Our model is inspired by that of Yilek for the RA setting: via access to an **Enc** oracle, we allow the adversary to get arbitrary messages

encrypted under arbitrary public keys, using functions ϕ of an initial set of well-distributed but unknown random values. The public keys can even be maliciously generated, and the adversary can of course know all the corresponding private keys. The adversary is tasked with winning an indistinguishability-style game, via an **LR** oracle which gives access to encryptions of left or right messages with respect to an honestly generated target public key pk^*, but again where the adversary can force the use of functions ϕ of the initial random values. When the functions ϕ are limited to coming from some set Φ, we speak of a Φ-restricted adversary.

Because the adversary may know all but one of the private keys, it can check that its challenger is behaving correctly with respect to its encryption queries. This also rules out the possibility of achieving RRA-ATK security for any randomness recovering PKE scheme, like RSA-OAEP [7] and PKE schemes based on the Fujisaki-Okamoto transformation [17]. Moreover, the encryption queries concern public keys that are outside the control of the challenger. This increases the technical challenge of achieving security in the RRA setting. This facet of the RRA setting bears comparison with the RKA setting for PKE [4, 6, 32]. In the RKA setting, the tampering via related key functions only affects the PKE scheme's private key, and so only comes into play when simulating *decryption* queries. By contrast, it is *encryption* queries that require special treatment in our RRA setting.

Given the power of the adversary in the RRA setting, we have to exclude certain sets of adversarial queries to prevent the adversary from trivially breaking security. For example, as in the RKA setting, constant functions ϕ must be disallowed for security to be achievable. See Section 2 for further discussion.

ROM construction We are able to show that, in the ROM, these necessary conditions on the function set Φ are actually also sufficient. More specifically, we show how to transform any IND-ATK secure PKE scheme `PKE` into a new PKE scheme `Hash-PKE` that is RRA-ATK secure, simply by hashing the random input together with the public key and message during encryption. In fact, this is just an application of the hedging approach from [27], and an instance of the randomized-encrypt-with-hash (REwH) scheme from [2]. Our result then shows that this approach also provides security in our new RRA setting.

Standard model constructions Having dealt with the ROM, we then turn our attention to constructions in the standard model. Reinforcing the connections to RKA security, we are able to show that any Φ-restricted RKA-PRF can be used to build a RRA-ATK secure PKE scheme for Φ-restricted adversaries, thus transferring security from the RKA setting (for PRFs) to the RRA setting for PKE. But the limited range of RKA-PRFs currently available in the literature [25, 3] essentially restricts the obtained RRA-ATK secure PKE scheme to a class of functions Φ consisting of *linear* or *group-induced* functions. To achieve an RRA-ATK secure PKE scheme for richer classes of functions, we must seek alternative methods of construction.

Unfortunately, we have not been able to achieve our full RRA-ATK security notion for more interesting function classes using other constructions. So we must resort to exploring alternative versions of this notion in order to make progress. We relax RRA-ATK security along two independent dimensions: the degree of control that the adversary enjoys over the public keys under which it can force encryptions for related random values, and the degree of adaptivity it has in the selection of functions $\phi \in \Phi$:

- We first consider the situation where the public keys are all honestly generated at the start of the security game, and the public keys and all but one of the private keys are then given to the adversary — the honest-key, related randomness attack (HK-RRA) setting. This is a reasonable relaxation in that, in practice, all the public keys that the adversary might be able to induce a user to encrypt under would be properly generated by users and then certified by a CA ahead of time. In this setting, we provide a generic construction for a scheme achieving HK-RRA-ATK security based on combining any IND-ATK secure PKE scheme with a Correlated-Input Secure (CIS) hash function [19]. Currently known instantiations of CIS hash functions allow us to obtain selective, HK-RRA-ATK security for Φ-restricted adversaries where Φ is a large class of polynomial functions (as opposed to the linear functions we can achieve using our RKA-PRF-based construction). Here, selectivity refers to the adversary committing at the start of the game to the set of functions it will use.
- We then consider the situation where there is no restriction on public keys, but the adversary is committed up-front to a vector of functions $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_q)$ that it will use in its attack, and where security is in the end quantified over all choices of $\boldsymbol{\phi}$ from some set $\boldsymbol{\Phi}$. This quantification is subtly different from allowing the adversary a fully adaptive choice of functions $\phi \in \Phi$ (for a detailed discussion, see Section 2). In this situation, we refer to the function-vector, related randomness attack (FV-RRA) model. Here, we are able to give a direct construction for a PKE scheme that is FV-RRA-ATK secure solely under the DDH assumption, assuming the component functions ϕ_i of $\boldsymbol{\phi}$ are simultaneously hard to invert on a random input. Our scheme is inspired by a PKE scheme of Boneh *et al.* [12] that is secure in the so-called *auxiliary input setting*, wherein the adversary is given a hard-to-invert function of the secret key as part of its input. By swapping the roles of secret key and randomness in the Boneh *et al.* scheme, we are able to obtain security in a setting where a hard-to-invert function of the encryption randomness is leaked to the adversary. This leakage is then sufficient to allow us to simulate the encryptions for adversarially chosen public keys. For technical reasons, to obtain a construction, we must also limit our adversary to using the identity function when accessing its **LR** oracle.

To summarise, in the standard model, we can achieve our full security notion, RRA-ATK security, but only for a limited class of functions Φ (inherited from known results on RKA-PRFs), while we can achieve alternative security notions for richer classes Φ.

1.3 Future Directions

In this paper, we concentrate on PKE, but RRA security notions can be developed for other primitives. As previously noted, the case of signatures is quite simple, provided one is prepared to extend a scheme's private key. We would expect symmetric key encryption and key exchange primitives to be more complex. Also as noted above, our RRA setting is related to the RKA setting, and it is an open problem to develop these connections further, possibly by considering a combined RKA/RRA setting.

2 Related Randomness Security for PKE

We now formalise our notions of related randomness security for PKE. We give a detailed treatment of our strongest notion, before sketching restricted versions. The description of our security notions will utilise code-based games and the associated language (see [8]).

Our strongest security notion, RRA-CCA security, is defined via the game in Figure 1. Here, a challenge key pair (pk^*, sk^*) for a PKE scheme $\mathtt{PKE} = (\mathtt{PKE.K}, \mathtt{PKE.E}, \mathtt{PKE.D})$ with randomness space $\mathtt{Rnd}$ is honestly generated, and the adversary is considered successful if it wins an indistinguishability game with respect to messages encrypted under pk^*. Extending the standard PKE setting, the adversary is able to control which one of polynomially many random values $r_i \in \mathtt{Rnd}$ is used in responding to each encryption query for pk^*; furthermore, the adversary is able to obtain the encryption of messages of its choice under (possibly maliciously generated) arbitrary public keys. Extending the model of Yilek [33], our adversary not only specifies which one of the random values r_i is to be used in each query, but also specifies, for each query he makes, a function ϕ on $\mathtt{Rnd}$; the value $\phi(r_i)$ is used for encryption in place of r_i. In the CCA setting, the adversary also has access to a regular decryption oracle for private key sk^*. Note that if the adversary uses *only* the identity function, then we recover the Resettability Attack (RA) model of Yilek [33].

It is not difficult to see that, as in the RA setting, an adversary may trivially win this game if no restrictions are placed on oracle queries.[1] We will shortly introduce an *equality-pattern respecting* definition for adversaries, designed to prevent trivial wins of this kind. This extends the related RA definition from [33]. However, restrictions on the functions ϕ will also be required. To illustrate the issue, consider as an extreme case the constant function ϕ_C (with $\phi_C(r) = C$ for all $r \in \mathtt{Rnd}$). Suppose the adversary submits **LR** query (m_0, m_1, j, ϕ_C) for any $m_0 \neq m_1$ and any $j \in \mathbb{N}$; the adversary receives a ciphertext c^* and then

[1] For example, if an adversary requests the encryption of m under the target public key using coins $\phi(r_i)$, $\mathtt{PKE.E}(pk^*, m; \phi(r_i))$, and submits **LR** query (m, m', i, ϕ), then the adversary guesses b is 0 if the two ciphertexts match, otherwise he guesses b is 1. This adversary wins the game with probability 1. As in the RA setting, such wins are unavoidable in our setting since encryption essentially becomes deterministic when the same random coins and functions ϕ are used.

proc. Initialise(λ):
$b \leftarrow_{\$} \{0,1\}$;
$(pk^*, sk^*) \leftarrow_{\$} \mathtt{PKE.K}(1^\lambda)$;
CoinTab $\leftarrow \emptyset$;
$\mathcal{S} \leftarrow \emptyset$; Return pk^*

proc. Dec(c):
If $c \in \mathcal{S}$, then return $\perp$
Else return $\mathtt{PKE.D}(sk^*, c)$

proc. LR(m_0, m_1, i, ϕ):
If CoinTab[i] $=\perp$
  CoinTab[i] $\leftarrow_{\$}$ Rnd
$r_i \leftarrow$ CoinTab[i]
$c \leftarrow \mathtt{PKE.E}(pk^*, m_b; \phi(r_i))$
$\mathcal{S} \leftarrow \mathcal{S} \cup \{c\}$
Return c

proc. Enc(pk, m, i, ϕ):
If CoinTab[i] $=\perp$
  CoinTab[i] $\leftarrow_{\$}$ Rnd
$r_i \leftarrow$ CoinTab[i]
$c \leftarrow \mathtt{PKE.E}(pk, m; \phi(r_i))$
Return c

proc. Finalise(b'):
If $b = b'$, return 1

Fig. 1. Game RRA-ATK. (Note that if ATK = CPA, then the adversary's access to **proc. Dec** is removed.)

computes $c_0 = \mathtt{PKE.E}(pk^*, m_0; C)$; the adversary outputs guess $b' = 0$ if and only if $c^* = c_0$. It is easy to see that this adversary wins the RRA-ATK game with probability 1. This example is analogous to one in the related key attack setting for PRFs in [5]. Hence, we will need to restrict the class of functions which the adversary is allowed to access in its queries to come from some set Φ, in which case we speak of Φ-restricted adversaries. We have already seen that constant functions must be excluded from Φ if we are to have any hope of achieving our related randomness security notion.

Thus we have two sets of constraints that we need to consider to prevent trivial wins: those on messages and randomness indices (analogous to the RA setting from [33]) and those on functions ϕ (analogous to the RKA setting for PRFs from [5]). Let us deal with the first set of constraints first and define what it means for an adversary to be equality-pattern respecting. The following definition is adapted from [33] for our purposes.

Definition 1. *Let $\mathcal{A}$ be a Φ-restricted adversary in Game* RRA-ATK *that queries r different randomness indices to its **LR** and **Enc** oracles and makes $q_{i,\phi}$ queries to its **LR** oracle with index i and function $\phi \in \Phi$. Let $E_{i,\phi}$ be the set of all messages m such that $\mathcal{A}$ makes **Enc** query (pk^*, m, i, ϕ). Let $(m_0^{i,\phi,1}, m_1^{i,\phi,1}), \ldots, (m_0^{i,\phi,q_{i,\phi}}, m_1^{i,\phi,q_{i,\phi}})$ be $\mathcal{A}$'s **LR** queries for index $i \in [r]$ and $\phi \in \Phi$. Suppose that for all pairs $(i,\phi) \in [r] \times \Phi$ and for all $j \neq k \in [q_{i,\phi}]$, we have:*

$$m_0^{i,\phi,j} = m_0^{i,\phi,k} \text{ iff } m_1^{i,\phi,j} = m_1^{i,\phi,k}$$

and that, for all pairs $(i,\phi) \in [r] \times \Phi$, and for all $j \in [q_{i,\phi}]$, we have:

$$m_0^{i,\phi,j} \notin E_{i,\phi} \wedge m_1^{i,\phi,j} \notin E_{i,\phi}.$$

Then we say that $\mathcal{A}$ is equality-pattern respecting.

Notice that if the adversary is restricted to using only the identity function, then this definition reduces to the equality-pattern respecting definition for the RA setting, cf. [33, Appendix A].

Definition 2. *We define the advantage of an equality-pattern respecting,* RRA-ATK *adversary* $\mathcal{A}$ *against a PKE scheme* PKE *to be:*

$$\mathbf{Adv}^{\text{rra-atk}}_{\mathtt{PKE},\mathcal{A}}(\lambda) := 2 \cdot \mathbb{P}[\text{RRA-ATK}^{\mathcal{A}}_{\mathtt{PKE}}(\lambda) \Rightarrow 1] - 1.$$

A PKE scheme PKE *is said to be* Φ-RRA-ATK *secure if the advantage of any* Φ*-restricted, equality-pattern respecting,* RRA-ATK *adversary against* PKE *that runs in polynomial time is negligible in the security parameter* λ*.*

2.1 Alternative security notions

The above definition for Φ-RRA-ATK security is very powerful: it allows an adversary to submit *any* public key to its encryption oracle and allows the adversary to *adaptively* choose the functions ϕ, the only restriction being that they lie in Φ. In Section 2.2 we will exhibit conditions that are both necessary and sufficient for achieving security in this sense in the ROM (given a starting PKE scheme that satisfies the usual definition of IND-ATK security). In the standard model, we will give a construction that relies on RKA-PRFs. Since constructions for these are currently very limited in terms of the function classes they can handle, we will now consider alternative versions of the Φ-RRA-ATK notion.

The first alternative notion we consider is called *Honest Key Related Randomness* (HK-RRA) security. The security game has two parameters, λ and ℓ. Informally, the game itself generates a polynomial number ℓ of key pairs and returns the public keys to the adversary. The adversary then chooses which public key he wishes to be the target key, and is given the private keys corresponding to all the non-target public keys. Meanwhile, the adversary's queries to its **Enc** oracle are restricted to using the public keys generated by the game. Suitable Φ-HK-RRA-ATK security notions follow by analogy with our earlier definitions.

One may consider notions intermediate between Φ-RRA-ATK security and Φ-HK-RRA-ATK security. For example, a registered key notion could be defined, in which the adversary chooses and registers key pairs (pk, sk), with registration involving a test for validity by some procedure, and all queries involve only registered public keys. One may also consider weaker variants of these notions in which the adversary's choice of functions ϕ is non-adaptive (or *selective*). That is, the adversary must submit a set of functions $\{\phi\} \subset \Phi$ of polynomial size to the game before he is allowed to see the target public key (or set of public keys, if playing in the Honest Key setting). In this setting, we refer to Φ-sHK-RRA-ATK security.

The final alternative notion we consider is called *Function-Vector Related Randomness* (FV-RRA) security, and is based on the game in Figure 2. Here, the adversary is parameterised by a vector of functions $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_q)$, and is limited to using only these functions in its oracle queries. Additionally, we restrict the adversary by demanding that the **LR** queries use only the identity function. However, once again, the adversary has complete freedom over public keys submitted to its encryption oracle. Furthermore, security will be quantified over *all* choices of vector from a particular class. (Specifically, in our construction

proc. Initialise(λ):
$b \leftarrow_{\$} \{0,1\}$;
$(pk^*, sk^*) \leftarrow_{\$} \texttt{PKE.K}(1^\lambda)$;
CoinTab $\leftarrow \emptyset$; $\mathcal{S} \leftarrow \emptyset$;
return pk^*

proc. Dec(c):
If $c \in \mathcal{S}$, then return $\perp$
Else return $\texttt{PKE.D}(sk^*, c)$

proc. LR(m_0, m_1, i):
If CoinTab[i] $=\perp$,
  CoinTab[i] $\leftarrow_{\$}$ Rnd
$r_i \leftarrow$ CoinTab[i]
$c \leftarrow \texttt{PKE.E}(pk^*, m_b; r_i)$
$\mathcal{S} \leftarrow \mathcal{S} \cup \{c\}$
return c

proc. Enc(pk, m, i, j):
If CoinTab[i] $=\perp$,
  CoinTab[i] $\leftarrow_{\$}$ Rnd
$r_i \leftarrow$ CoinTab[i]
$c \leftarrow \texttt{PKE.E}(pk, m; \phi_j(r_i))$
return c

proc. Finalise(b'):
If $b = b'$, return 1

Fig. 2. Game $\boldsymbol{\phi}$-FV-RRA-ATK, where $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_q)$. (As usual, if ATK = CPA, then the adversary's access to **proc. Dec** is removed.)

in Section 5, we will demand that security holds over all vectors $\boldsymbol{\phi}$ that are simultaneously hard to invert on a common random input r.) This quantification actually makes our notion rather strong.

Definition 3. *Let $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_q)$ be a vector of $q := q(\lambda)$ functions. We define the advantage of an equality-pattern respecting, $\boldsymbol{\phi}$-*FV-RRA-ATK *adversary $\mathcal{A}$ against a PKE scheme* PKE *to be:*

$$\mathbf{Adv}^{\boldsymbol{\phi}\text{-fv-rra-atk}}_{\texttt{PKE},\mathcal{A}}(\lambda) := 2 \cdot \mathbb{P}[\boldsymbol{\phi}\text{-FV-RRA-ATK}^{\mathcal{A}}_{\texttt{PKE}}(\lambda) \Rightarrow 1] - 1.$$

If $\boldsymbol{\Phi}$ is a set of vectors of functions, then a PKE scheme PKE *is said to be $\boldsymbol{\Phi}$-*FV-RRA-ATK *secure if, for all $\boldsymbol{\phi} \in \boldsymbol{\Phi}$, the advantage of any equality-pattern respecting, $\boldsymbol{\phi}$-*FV-RRA-ATK *adversary against* PKE *that runs in polynomial time is negligible in the security parameter λ.*

Comparison of security notions The first alternative security notion, HK-RRA-ATK security, is easily seen to be a strictly weaker notion than full RRA-ATK security[2]. Likewise, the selective models are easily seen to be weaker then their adaptive counterparts. However, the relation between full RRA-ATK security and FV-RRA-ATK security is not immediately obvious. Aside from the restriction on **LR**-queries in FV-RRA-ATK security, there is a subtle distinction between requiring security for all vectors $\boldsymbol{\phi}$ of functions from a particular set Φ and requiring security for a fully adaptive choice of functions $\phi \in \Phi$. In particular, the former notion will allow a security reduction to consider multiple runs of an adversary with different random coins for a fixed choice of function vector $\boldsymbol{\phi}$, whereas the latter notion will leave open the possibility that an adversary will chose a different sequence of functions ϕ in each run. Also note that FV-RRA-ATK security guarantees that there is no choice of $\boldsymbol{\phi}$ for which the

[2] A separation can be established by considering a scheme where public keys generated by the key generation algorithm always have a certain bit set to 0, and where the encryption algorithm, given a public key with this bit set to 1 (i.e. a maliciously generated public key), will expose the randomness used for the encryption.

considered scheme is weak, even if this choice might be computationally hard for an adaptive adversary to find. Furthermore, the relation between the notions might also be influenced by the considered class of functions Φ. It remains future work to fully explore and categorise the possible notions of RRA security.

It is not hard to see that our RRA security notions are incomparable with the CDA security notions of [2]. In the RA setting, Yilek defines only an equivalent of our full RRA-ATK notion; it is clear that RRA-ATK security is stronger than his RA-ATK security whenever the function set Φ contains the identity function. The same would carry over to relaxed versions of RA-ATK security.

2.2 Function restrictions

Above, we briefly alluded to the fact that the class of functions Φ used by our RRA adversaries must be restricted in various ways. The example given showed that constant functions must always be excluded. Here, we exhibit much stronger necessary conditions on Φ that must be satisfied, namely output-unpredictability and collision-resistance. These notions are closely related to notions with the same names arising in the setting of related key security for PRFs that was considered in [5]. Here, however, we are concerned with functions acting on the randomness used in PKE schemes rather than on PRF keys.

Definition 4 (Output-unpredictability for Φ). *Let Φ be a set of functions from* `Rnd` *to* `Rnd`*. Let α and β be positive integers. Then the (α, β)-output-unpredictability of Φ is defined to be:*

$$\mathrm{InSec}^{\mathrm{up}}_{\Phi}(\alpha, \beta) = \max_{P \subseteq \Phi, X \subseteq \mathcal{R}, |P| \leq \alpha, |X| \leq \beta} \left\{ \mathbb{P}\left[r \leftarrow_{\$} \mathtt{Rnd} : \{\phi(r) : \phi \in P\} \cap X \neq \emptyset \right] \right\}.$$

Definition 5 (Collision-resistance for Φ). *Let Φ be a set of functions from* `Rnd` *to* `Rnd`*. Let α be a positive integer. Then the α-collision-resistance of Φ is defined to be:*

$$\mathrm{InSec}^{\mathrm{cr}}_{\Phi}(\alpha) = \max_{P \subseteq \Phi, |P| \leq \alpha} \left\{ \mathbb{P}\left[r \leftarrow_{\$} \mathtt{Rnd} : |\{\phi(r) : \phi \in P\}| < |P| \right] \right\}.$$

Regarding these two definitions, we have the two following results.

Theorem 1 (Necessity of output-unpredictability). *Let Φ be a class of functions from* `Rnd` *to* `Rnd`*. Suppose there are natural numbers $\alpha = \mathrm{poly}_1(\lambda)$ and $\beta = \mathrm{poly}_2(\lambda)$ such that $\mathrm{InSec}^{\mathrm{up}}_{\Phi}(\alpha, \beta) = p$, where $p := p(\lambda)$ is non-negligible. Then no* PKE *scheme can be* RRA-ATK *secure with respect to the class of functions Φ.*

Theorem 2 (Necessity of collision-resistance). *Let Φ be a class of functions from* `Rnd` *to* `Rnd`*. Suppose there is a natural number $\alpha = \mathrm{poly}_1(\lambda)$ such that $\mathrm{InSec}^{\mathrm{cr}}_{\Phi}(\alpha) = p$, where $p := p(\lambda)$ is non-negligible. Then no* PKE *scheme can be* RRA-ATK *secure with respect to the class of functions Φ.*

Alg. PRF-PKE.K(1^λ):	Alg. PRF-PKE.E(pk, m):	Alg. PRF-PKE.D(sk, c):
$(pk, sk) \leftarrow_\$ \texttt{PKE.K}(1^\lambda)$	$r \leftarrow_\$ \texttt{Rnd}$ $r' \leftarrow F_r(pk\|\|m)$ $c \leftarrow \texttt{PKE.E}(pk, m; r')$ return c	$m \leftarrow \texttt{PKE.D}(sk, c)$ return m

Fig. 3. Scheme PRF-PKE built from a standard PKE scheme, PKE and a PRF, F.

We note that many classes of functions that arise from practical attacks satisfy these conditions. For example, the class of functions that flip bits at certain positions, or the class of functions that fix the value of certain bits, are both output-unpredictable and collision-resistant (provided at least a polynomial number of bits are not fixed, in the latter case).

In the RO model, these conditions are sufficient to achieve security in our strongest randomness attacks. More specifically, we can transform any IND-ATK secure scheme into a RRA-ATK secure scheme, simply by hashing string representations of the public key, the message, and appropriate randomness, and then using the output as randomness for the standard encryption scheme. This is an instance of the randomized-encrypt-with-hash (REwH) scheme from [2]. If the class of functions Φ is sufficiently collision-resistant and output-unpredictable, then this scheme is RRA-ATK-secure. We defer the details to the full version.

3 Related Randomness Security from RKA-PRFs

Since the RA setting of [33] is a special case of our RRA setting, an obvious way to try to achieve RRA security is to extend the main construction from [33]. That construction combines a PRF with an IND-ATK secure PKE scheme. Specifically, the randomness r is used as a key to the PRF, and the input to the PRF is the "context" $pk||m$; the output from the PRF is then used as the actual randomness for encryption. This construction extends directly to our setting, and security is guaranteed against Φ-restricted adversaries in our strongest RRA-ATK models, under the assumption that the PRF is Φ-RKA-secure (i.e. secure against related key attacks for the *same* class of functions Φ). Thus the construction transfers RKA security for PRFs to RRA-ATK security for PKE. Figure 3 formalises the construction, and Theorem 3 our security result.

Theorem 3. *Suppose $\mathcal{A}$ is a Φ-restricted, equality-pattern respecting adversary in the* RRA-ATK *game against the scheme* PRF-PKE *defined in Figure 3. Suppose $\mathcal{A}$ makes q_{LR} **LR** queries, q_s **Enc** queries, and uses q_r randomness indices. Then there exists a Φ-restricted RKA-PRF adversary $\mathcal{B}$ and an IND-ATK adversary $\mathcal{C}$ such that*

$$\mathbf{Adv}^{\text{rra-atk}}_{\texttt{PRF-PKE},\mathcal{A}}(\lambda) \leq q_{LR} \cdot q_r \cdot \mathbf{Adv}^{\text{ind-atk}}_{\texttt{PKE},\mathcal{C}}(\lambda) + 2q_r \cdot \mathbf{Adv}^{\text{rka-prf}}_{F,\mathcal{B}}(\lambda).$$

*Adversaries $\mathcal{B}$ and $\mathcal{C}$ run in approximately the same time as $\mathcal{A}$. Adversary $\mathcal{C}$ makes 1 **LR** query and the same number of **Dec** queries as $\mathcal{A}$. Adversary $\mathcal{B}$ makes at most $q_{LR} + s$ queries to its oracle.*

Notice that our RO scheme (mentioned in Section 2) may be interpreted as an instantiation of our scheme in Figure 3, since a random oracle can be viewed as an (unkeyed) RKA-PRF.

The previous theorem is seductively simple, but currently of limited application because the set of known RKA-secure PRFs is rather sparse. RKA-PRFs were first formalised in 2003 by Bellare and Kohno [5], and some initial (though not fully satisfactory) constructions were given in [5] and [25]. Setting these aside, the only known constructions are due to Bellare and Cash [3]. They gave a first construction for an RKA-PRF (based on the Naor-Reingold PRF) which is provably secure under the DDH assumption for related key functions Φ corresponding to component-wise multiplication on the key-space $(\mathbb{Z}_p^*)^{n+1}$. They also provided a second construction achieving a similar result under the DLIN assumption. A third construction for related key functions Φ corresponding to component-wise *addition* on the key-space $(\mathbb{Z}_p)^n$ was recently withdrawn by the authors of [3].

The limited nature of existing RKA-PRF families forces us to find alternative approaches to achieving security in the RRA setting. The application for RKA-PRFs implied by Theorem 3 also provides yet more motivation for the fundamental problem of constructing RKA-PRFs for richer classes of related key function.

4 Related Randomness PKE from CIS Hash Functions

To address some of the limitations encountered in the previous approach, we show how a PKE scheme secure in the RRA setting can be constructed using correlated-input secure (CIS) hash functions as introduced in [19]. While the currently known instantiations of CIS hash functions only allow us to obtain selective HK-RRA-ATK security, we are able to obtain security for a large class of polynomial functions, as opposed to linear functions to which the previous construction is currently restricted.

In its strongest form, a CIS hash function h (with key k) will yield output $h_k(x)$ which is pseudorandom, even when given the hash value of multiple correlated input values $(h_k(\phi_1(x)), \ldots, h_k(\phi_q(x)))$, where the correlation functions $\phi_1, \ldots, \phi_q$ are maliciously chosen. This type of CIS hash function is closely related to RKA-secure PRFs. In fact, the authors of [19] show that given a CIS hash function h, an RKA-secure *weak* PRF F can be obtained simply by exchanging the role of the key and the input of h:

$$F_K(x) := h_x(K).$$

Recall that weak PRF security does not allow an adversary to choose the function inputs, but instead, the inputs are chosen uniformly at random in the security game.

Alg. $\texttt{CI-Hash-PKE.K}(1^\lambda)$:	Alg. $\texttt{CI-Hash-PKE.E}(\hat{pk}, m)$:	Alg. $\texttt{CI-Hash-PKE.D}(\hat{sk}, c)$:
$(pk, sk) \leftarrow_\$ \texttt{PKE.K}(1^\lambda)$	$(pk\|\|k) \leftarrow \hat{pk}$	$m \leftarrow \texttt{PKE.D}(\hat{sk}, c)$
$k \leftarrow_\$ \texttt{CI-HASH.K}(1^\lambda)$	$r \leftarrow_\$ \texttt{Rnd}$	return m
$(\hat{pk}, \hat{sk}) \leftarrow (pk\|\|k, sk)$	$r' \leftarrow h_k(r)$	
	$r'' \leftarrow F_{r'}(\hat{pk}\|\|m)$	
	$c \leftarrow \texttt{PKE.E}(pk, m; r'')$	
	return c	

Fig. 4. Scheme CI-Hash-PKE built from PKE scheme PKE, PRF F, and hash function family $\mathcal{H}$.

The authors of [19] furthermore give a concrete construction of a CIS hash function secure for a class of correlation functions consisting of uniform-output[3] polynomials of bounded degree, albeit in a restricted security model where the adversary's function queries are non-adaptive. This then yields a non-adaptive, RKA-secure weak PRF.

Unfortunately, such a PRF this is not sufficient for our purposes. Surprisingly, however, by making a relatively simple modification to the above construction of PRFs from CIS hash functions, it is possible to obtain a primitive similar to an RKA-secure (standard) PRF. More specifically, consider a CIS hash function h and a standard PRF f. We introduce a public parameter c of F which will correspond to the key for h, and then, instead of using the output of h directly, we use h to derive a key for f. More specifically, we define

$$F_{c,K}(x) := f_{h_c(K)}(x).$$

Whilst not strictly an RKA-secure PRF due to the presence of the public parameter c, this primitive allows adaptively chosen inputs x, while remaining secure under related key attacks. This 'partial' RKA-secure PRF will allow us to obtain HK-RRA-ATK secure encryption schemes for the function families of the underlying CIS hash function h. However, to achieve this, we need to extend the definitions and theorems of [19] to the multi-key setting (reflecting the fact that in the HK-RRA setting, our adversary can interact with multiple public keys). The extensions of the security definitions are relatively straightforward, and we defer definitions of a *multi-key selective correlated-input pseudorandom* (MK-SCI-PR) secure family of hash functions to the full version.

Based on an ordinary PKE scheme PKE, a PRF F, and a family of hash functions $\mathcal{H}$, we construct a PKE scheme CI-Hash-PKE as shown in Figure 4. The following theorem establishes the selective ℓ-HK-RRA-ATK security of this scheme based on the IND-ATK security of PKE, the multi-key selective CIS security of $\mathcal{H}$, and the (regular) pseudorandomness of F.

Theorem 4. *Suppose $\mathcal{A}$ is a Φ-restricted, equality pattern respecting adversary in the selective* ℓ-HK-RRA-ATK *game against the scheme* CI-Hash-PKE *in Fig-*

[3] A polynomial is said to be a uniform-output polynomial if its output range is equal to its domain i.e. evaluating the polynomial on all values in the domain will again yield the elements of the domain.

*ure 4. Suppose $\mathcal{A}$ makes q_{LR} **LR** queries, uses q_r randomness indices, and uses q_ϕ functions in its oracle queries. Then there exists a Φ-restricted, multi-key, selective correlated-input hash adversary $\mathcal{B}$, a* PRF *adversary $\mathcal{C}$ and an* IND-ATK *adversary $\mathcal{D}$ such that*

$$\mathbf{Adv}^{\ell\text{-shk-rra-atk}}_{\texttt{CI-Hash-PKE},\mathcal{A}}(\lambda) \leq 2q_\phi \cdot q_r \cdot \mathbf{Adv}^{\ell\text{-mk-sci-pr}}_{\mathcal{H},\mathcal{B}}(\lambda) + 2q_\phi \cdot q_r \cdot \mathbf{Adv}^{\text{prf}}_{F,\mathcal{C}}(\lambda) + \ell \cdot q_{LR} \cdot q_r \cdot \mathbf{Adv}^{\text{ind-atk}}_{\texttt{PKE},\mathcal{D}}(\lambda) + \frac{\ell^2 \cdot q_r}{|\text{HashKeySpace}|}.$$

Adversaries $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ run in approximately the same time as $\mathcal{A}$. Adversary $\mathcal{C}$ makes at most q_{LR} queries, and $\mathcal{D}$ makes 1 ***LR*** *query and as many* ***Dec*** *queries as $\mathcal{A}$.*

It remains to show that we can instantiate a hash function satisfying the multi-key correlated-input security notion. We achieve this by extending the security results for the CIS hash function defined in [19]. Concretely, the CIS hash function from [19] is defined as follows:

$\texttt{GenFun}(1^\lambda)$: Pick a group $\mathbb{G}$ of prime order p, and set the keyspace to $\mathcal{K} = \mathbb{G} \times \mathbb{Z}_p$, the domain to $\mathcal{D} = \mathbb{Z}_p$, and the range to $\mathcal{R} = \mathbb{G}$. Return $(\mathcal{K}, \mathcal{D}, \mathcal{R}, h)$ where h is a description of the function defined below.

$h_k(x)$: For $k \in \mathcal{K}$ and $x \in \mathcal{D}$, parse k as $(g, a) \in \mathbb{G} \times \mathbb{Z}_p$ and return

$$h_k(x) = g^{\frac{1}{x+a}},$$

where $1/(m+a)$ is computed modulo p

Based on the decisional q-Diffie Hellman Inversion (q-DDHI) assumption in $\mathbb{G}$, and extending the results of [19], we are able to show that the above hash function achieves multi-key correlated-input pseudorandomness for a class of functions consisting of uniform-output polynomials of bounded degree.

Theorem 5. *Assume the decisional q-DDHI assumption holds in $\mathbb{G}$, and let Φ be a class of uniform-output polynomials over $\mathbb{Z}_p$. Then there exists no polynomial time Φ-restricted adversary $\mathcal{A}$ with non-negligible advantage in the* (Φ, ℓ)-MK-SCI-PR *security game when interacting with $\mathcal{H}$ defined as above, provided that $\ell \cdot d \leq q+1$, where d is an upper bound on the sum of the degrees of the polynomials submitted by $\mathcal{A}$. More precisely, if $\ell \cdot d \leq q+1$, then for any polynomial time Φ-restricted $\mathcal{A}$, there exists a polynomial time algorithm $\mathcal{B}$ such that*

$$\mathbf{Adv}^{\ell\text{-mk-sci-pr}}_{\mathcal{H},\mathcal{A}}(\lambda) \leq 2n\ell \cdot \mathbf{Adv}^{q\text{-ddhi}}_{\mathbb{G},\mathcal{B}}(\lambda)$$

where n is the number of polynomials submitted by $\mathcal{A}$.

Note 1. Our 'partial' RKA-secure PRF is only secure when an adversary's function queries are non-adaptive, which is why we are only able to prove selective HK-RRA-ATK security. If we had a result similar to Theorem 5 for adaptive function queries, then we would immediately obtain a PKE scheme that is (adaptively) HK-RRA-ATK secure.

Note 2. The above construction is only shown to achieve HK-RRA-ATK security, as opposed to RRA-ATK security. The technical reason for this is that public keys include a hash key, and the CIS hash function is only assumed to be secure for honestly generated keys. An alternative solution would be to introduce a *common reference string* (CRS) containing a single hash key, and let all users make use of this. While this requires a trusted third party to initially set up the CRS, it would be possible to show RRA-ATK security of the above construction in a security model appropriately extended to model the presence of a CRS.

Likewise, if we had a multi-key CIS hash function that remained secure for maliciously chosen keys, then we would be able to obtain full RRA-ATK security for the above construction. Unfortunately, we are currently unaware of how to obtain such CIS hash functions.

5 Function-Vector Related Randomness Security

Our previous standard model constructions concerned functions ϕ that are linear (scheme `PRF-PKE` analysed in Theorem 3 combined with known RKA-PRF families), or of bounded degree and having unpredictable outputs (scheme `CI-Hash-PKE` analysed in Theorem 5). We now turn our attention to alternative classes of functions. Specifically, we will propose a construction for a PKE scheme that is $\boldsymbol{\Phi}$-FV-RRA-ATK secure for the set $\boldsymbol{\Phi}$ of vectors of functions that are hard to invert, in a sense that we make precise next.

Definition 6. *Let $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_q)$ denote a vector of functions on a set $\mathtt{Rnd}_\lambda$, where $q := q(\lambda)$ is polynomial in the security parameter λ. Let $\delta(\lambda)$ be a function. We say that $\boldsymbol{\phi}$ is* $\delta(\lambda)$-hard-to-invert *if, for all polynomial time algorithms $\mathcal{A}$ and all sufficiently large λ, we have:*

$$\mathbb{P}[r \leftarrow \mathcal{A}(\phi_1(r), \ldots, \phi_q(r)) : r \leftarrow_{\$} \mathtt{Rnd}_\lambda] \leq \delta(\lambda).$$

We say that a set of vectors of functions $\boldsymbol{\Phi}$ is δ-hard-to-invert if each vector $\boldsymbol{\phi} \in \boldsymbol{\Phi}$ is δ-hard-to-invert (note that the vectors in such a set $\boldsymbol{\Phi}$ need not all be of the same dimension, but we assume they each have dimension that is polynomial in λ).

We will now construct a PKE scheme that offers $\boldsymbol{\Phi}$-FV-RRA-CPA security, where $\boldsymbol{\Phi}$ is the set of *all* sufficiently hard-to-invert vectors of functions on the scheme's randomness space `Rnd`. As noted in Section 2, security in this setting is quantified over *all* vectors in $\boldsymbol{\Phi}$, and the adversary is allowed to work with any set of public keys (even maliciously generated) in its attack. This makes our result relatively strong.

With these definitions in hand, Figure 5 defines our PKE scheme `mBHHO` which offers security in the FV-RRA-CPA setting. This scheme is obtained by modifying a PKE scheme of Boneh *et al.* [12] (the `BHHO` scheme) which Dodis *et al.* [14] showed to be secure in the auxiliary input setting. To arrive at our modified scheme `mBHHO`, we swap the roles of secret key and randomness in

Alg. $\texttt{mBHHO.K}(1^\lambda)$:
$g_1, \ldots, g_\lambda \leftarrow_\$ \mathbb{G}$
$x \leftarrow_\$ \mathbb{Z}_p$
$pk = (g_1, \ldots, g_\lambda, g_1^x \ldots, g_\lambda^x)$
$sk = x$

Alg. $\texttt{mBHHO.E}(pk, m)$:
$r \leftarrow_\$ \{0,1\}^\lambda$
$c_1 = \prod_{i=1}^{\lambda} g_i^{r_i}$
$(K, r') \leftarrow f(\prod_{i=1}^{\lambda} (g_i^x)^{r_i})$
$r'' \leftarrow F_{r'}(pk||m)$
$c_2 = \texttt{DEM.E}(K, m; r'')$
$c = (c_1, c_2)$

Alg. $\texttt{mBHHO.D}(sk, (c_1, c_2))$:
$(K, r') \leftarrow f(c_1^x)$
$m \leftarrow \texttt{DEM.D}(K, c_2)$

Fig. 5. Modified BHHO scheme mBHHO, constructed using a PRF, F, a KDF, f, and a DEM DEM.

the original BHHO scheme. This then enables us to provide the values $\phi_i(r)$ as auxiliary inputs without undermining the usual IND-CPA security of the scheme; in turn, these values enables our security reduction to properly handle **Enc** queries involving any function ϕ_i. The following theorem gives our formal result concerning the security of this scheme.

Theorem 6. *Let $\boldsymbol{\Phi}$ be the set of δ-hard-to-invert vectors of functions on $\{0,1\}^\lambda$. The PKE scheme* mBHHO *in Figure 5 is $\boldsymbol{\Phi}$-*FV-RRA-CPA *secure. More precisely, consider any polynomial-size vector of functions $\boldsymbol{\phi} \in \boldsymbol{\Phi}$ and any equality-pattern respecting, $\boldsymbol{\phi}$-*FV-RRA-CPA *adversary $\mathcal{A}$ against* mBHHO*. Suppose $\mathcal{A}$ makes q_{LR} **LR** queries and uses q_r randomness indices. Then there exists a DDH adversary $\mathcal{B}$, a KDF adversary $\mathcal{D}$, a PRF adversary $\mathcal{E}$, and an IND-CPA adversary $\mathcal{F}$, all running in polynomial time, such that:*

$$\begin{aligned}\mathbf{Adv}^{\boldsymbol{\phi}\text{-fv-rra-cpa}}_{\texttt{mBHHO},\mathcal{A}}(\lambda) < {} & 2\lambda q_r \cdot \mathbf{Adv}^{\text{ddh}}_{\mathbb{G},\mathcal{B}}(\lambda) + 2q_r \cdot \mathbf{Adv}^{\text{kdf}}_{f,\mathcal{D}}(\lambda) \\ & + 2q_r \cdot \mathbf{Adv}^{\text{prf}}_{F,\mathcal{E}}(\lambda) + q_r \cdot \mathbf{Adv}^{\text{ind-cpa}}_{\texttt{DEM},\mathcal{F}}(\lambda) \\ & + q_r p^2 \sqrt[3]{512\lambda\delta}.\end{aligned}$$

In particular, when δ is sufficiently small the advantage of $\mathcal{A}$ is negligible in the security parameter λ.

The class of related randomness functions which our scheme mBHHO can tolerate is quite different from those in our previous constructions: linear and bounded-degree polynomials are certainly not hard-to-invert in general. Our proof of Theorem 6 actually shows that even if $\boldsymbol{\phi}(r)$ were to completely leak to the adversary (instead of merely being indirectly accessible via **Enc** queries), the scheme mBHHO would still be secure. This would not be the case if the analogous $\phi(r)$ values were to leak in our earlier schemes PRF-PKE and CI-Hash-PKE, since the adversary could actually reconstruct r from this leakage for the relevant ϕ functions and win the security game. Furthermore, the functions are not required to be collision-resistant or output-unpredictable. These restrictions are only strictly required of the functions queried to the **LR** oracle. However, since an an adversary is restricted to using only the identity function (which *is* collision-resistant and output-unpredictable) in its **LR** queries, the functions in $\boldsymbol{\Phi}$ do not need to satisfy these conditions.

References

1. Andrew Becherer, Alex Stamos, and Nathan Wilcox. Cloud computing security: Raining on the trendy new parade. *BlackHat USA*, 2009.
2. Mihir Bellare, Zvika Brakerski, Moni Naor, Thomas Ristenpart, Gil Segev, Hovav Shacham, and Scott Yilek. Hedged public-key encryption: How to protect against bad randomness. In Mitsuru Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 232–249. Springer, 2009.
3. Mihir Bellare and David Cash. Pseudorandom functions and permutations provably secure against related-key attacks. In Tal Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 666–684. Springer, 2010.
4. Mihir Bellare, David Cash, and Rachel Miller. Cryptography secure against related-key attacks and tampering. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT*, volume 7073 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2011.
5. Mihir Bellare and Tadayoshi Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In Eli Biham, editor, *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 491–506. Springer, 2003.
6. Mihir Bellare, Kenneth G. Paterson, and Susan Thomson. RKA security beyond the linear barrier: IBE, encryption and signatures. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 331–348. Springer, 2012.
7. Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De Santis, editor, *EUROCRYPT*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111. Springer, 1994.
8. Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Vaudenay [31], pages 409–426.
9. Mike Bendel. Hackers describe PS3 security as epic fail, gain unrestricted access, 2011. `http://www.exophase.com/20540/hackers-describe-ps3-security-as-epic-fail-gain-unrestricted-access/`.
10. Daniel J. Bernstein, Yun-An Chang, Chen-Mou Cheng, Li-Ping Chou, Nadia Heninger, Tanja Lange, and Nicko van Someren. Factoring RSA keys from certified smart cards: Coppersmith in the wild. Cryptology ePrint Archive, Report 2013/599, 2013. `http://eprint.iacr.org/`.
11. Bitcoin.org. Android security vulnerability, 2013. `http://bitcoin.org/en/alert/2013-08-11-android`.
12. Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In David Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 108–125. Springer, 2008.
13. Debian. Debian Security Advisory DSA-1571-1: OpenSSL – predictable random number generator, 2008. `http://www.debian.org/security/2008/dsa-1571`.
14. Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In Daniele Micciancio, editor, *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 361–381. Springer, 2010.
15. Yevgeniy Dodis, David Pointcheval, Sylvain Ruhault, Damien Vergnaud, and Daniel Wichs. Security analysis of pseudo-random number generators with input: /dev/random is not robust. *IACR Cryptology ePrint Archive*, 2013:338, 2013.
16. Leo Dorrendorf, Zvi Gutterman, and Benny Pinkas. Cryptanalysis of the random number generator of the Windows operating system. *ACM Trans. Inf. Syst. Secur.*, 13(1), 2009.

17. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554. Springer, 1999.
18. Ian Goldberg and David Wagner. Randomness and the Netscape browser, 1996. `http://www.drdobbs.com/windows/184409807`.
19. Vipul Goyal, Adam O'Neill, and Vanishree Rao. Correlated-input secure hash functions. In Yuval Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 182–200. Springer, 2011.
20. Zvi Gutterman and Dahlia Malkhi. Hold your sessions: An attack on java session-id generation. In Alfred Menezes, editor, *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 44–57. Springer, 2005.
21. Zvi Gutterman, Benny Pinkas, and Tzachy Reinman. Analysis of the linux random number generator. In *IEEE Symposium on Security and Privacy*, pages 371–385. IEEE Computer Society, 2006.
22. Nadia Heninger, Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. Mining your Ps and Qs: Detection of widespread weak keys in network devices. In *Proceedings of the 21st USENIX Security Symposium*, August 2012.
23. Seny Kamara and Jonathan Katz. How to encrypt with a malicious random number generator. In Kaisa Nyberg, editor, *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pages 303–315. Springer, 2008.
24. Arjen K. Lenstra, James P. Hughes, Maxime Augier, Joppe W. Bos, Thorsten Kleinjung, and Christophe Wachter. Public keys. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 626–642. Springer, 2012.
25. Stefan Lucks. Ciphers secure against related-key attacks. In Roy and Meier [30], pages 359–370.
26. Kai Michaelis, Christopher Meyer, and Jörg Schwenk. Randomly failed! the state of randomness in current java implementations. In Ed Dawson, editor, *CT-RSA*, volume 7779 of *Lecture Notes in Computer Science*, pages 129–144. Springer, 2013.
27. Thomas Ristenpart and Scott Yilek. When good randomness goes bad: Virtual machine reset vulnerabilities and hedging deployed cryptography. In *NDSS*. The Internet Society, 2010.
28. Phillip Rogaway. Nonce-based symmetric encryption. In Roy and Meier [30], pages 348–359.
29. Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In Vaudenay [31], pages 373–390.
30. Bimal K. Roy and Willi Meier, editors. *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, volume 3017 of *Lecture Notes in Computer Science*. Springer, 2004.
31. Serge Vaudenay, editor. *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*. Springer, 2006.
32. Hoeteck Wee. Public key encryption against related key attacks. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 262–279. Springer, 2012.
33. Scott Yilek. Resettable public-key encryption: How to encrypt on a virtual machine. In Josef Pieprzyk, editor, *CT-RSA*, volume 5985 of *Lecture Notes in Computer Science*, pages 41–56. Springer, 2010.